Mitigating Reply Implosions in Query-Based Service Discovery Protocols for Mobile Wireless Ad Hoc Networks

Antônio Tadeu Gomez, Artur Ziviani, Luciana Lima, Markus Endler, Guillaume Chelius

Mitigating Reply Implosions in Query-Based Service Discovery Protocols for Mobile Wireless Ad Hoc Networks*

Antônio Tadeu A. Gomes[1], Artur Ziviani[1], Luciana S. Lima[1,2], Markus Endler[2], and Guillaume Chelius[3]

[1] National Laboratory for Scientific Computing (LNCC)
Av. Getúlio Vargas 333, 25651-075, Petrópolis-RJ, Brazil
{atagomes,ziviani,lslima}@lncc.br
[2] Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente 225, 22453-900, Rio de Janeiro-RJ, Brazil
{lslima,markus}@inf.puc-rio.br
[3] INRIA ARES team – CITI Lab – INSA-Lyon
Villeurbanne, France
guillaume.chelius@inria.fr

Abstract. Providing service discovery in an efficient and scalable way in ad hoc networks is a challenging problem, in particular for multihop scenarios, due to the large number of potential participant nodes and the scarce resources in these networks. In this paper, we propose and evaluate an approach to mitigate the reply implosion problem in query-based service discovery protocols for multihop mobile ad hoc networks. Our simulation results show the scalability and efficiency of the proposed solution. We demonstrate that the proposed scheme considerably reduces the number of transmissions without compromising the efficiency of the service discovery in scenarios of pedestrian mobility.

1 Introduction

Efficient discovery of services, or resources, in arbitrary and ever-changing, dynamic network topologies is a key requirement of several distributed applications, such as grids with mobile nodes, P2P computing, or sensor networks. Nevertheless, research related to service discovery protocols (SDPs) in mobile ad hoc networks (MANETs) is relatively new—as compared with wired and infrastructure wireless networks [1]—and particularly challenging in multihop scenarios, as they are formed opportunistically and can change rapidly according to node mobility. Some approaches to service discovery in multihop MANETs incorporate the discovery functionality into the ad hoc routing protocols at the network and

* This work was supported by the Brazilian Funding Agencies FAPERJ, CNPq, and CAPES, and by the Brazilian Ministry of Science and Technology (MCT).

D. Coudert et al. (Eds.): ADHOC-NOW 2008, LNCS 5198, pp. 29–42, 2008.

link levels [2], but the inherent instability of such networks makes routing consistency hard to achieve, leading to inefficiency in service selection. Application-level SDPs—*i.e.* independent of the underlying ad hoc routing protocols—have also been proposed for such networks [3]. As usual, these protocols adopt one of the two basic approaches to exchange service information [4]: *service queries* and *service announcements*.[1] Both approaches raise issues when considered from the viewpoint of multihop MANETs. One the one hand, announcement-based protocols are clearly inadequate for computational resources (*e.g.* CPU load and available memory), such as the ones provided in mobile grids [5], because resource announcements would need to be constantly updated/refreshed with the current status of resource availability due to the dynamic nature of these resources, as their availability can considerably vary in short periods. On the other hand, query-based protocols can cause a serious waste of resources if consumer nodes naively flood much service requests over the network (a.k.a. the *broadcast storm problem*) and provider nodes naively reply to these requests (a.k.a. the *reply implosion problem*). As we are interested in dealing with dynamic resources, we focus on enabling more efficient query-based SDPs for multihop MANETs.

In this paper, we present a mechanism to relieve the reply implosion problem in query-based SDPs. The proposed *Suppression by Vicinity* (SbV) mechanism works in a peer-to-peer fashion, regardless of the underlying routing protocol and network-level addressing adopted in the MANET. The SbV mechanism assumes a service-usage model in which one or more service-providing nodes can reply to the same request and a consumer node can select one or more instances of the required service. Hence, different query-based SDPs can employ the SbV mechanism, with only minor adaptations.

To experiment with the SbV mechanism, we have incorporated it into the P2PDP protocol [6], a purely query-based SDP tailored for discovery of computational services in (single-hop) ad hoc mobile grids. The P2PDP protocol allows the simultaneous selection of multiple nodes as the most suitable providers—based on the availability of the specific resources being requested—of a particular computational service. We demonstrate through simulations that the use of the SbV mechanism improves the scalability of query-based SDPs in multihop MANETs. Our simulation results also show that the SbV mechanism reduces the overall network load generated by such protocols in a distributed way through the MANET. Moreover, these results indicate that the SbV mechanism does not compromise the efficiency of service discovery in the P2PDP protocol under scenarios of slow mobility, *i.e.* pedestrian walking speed.

The remainder of the paper is structured as follows. In Section 2 we survey some related work on service discovery protocols. We describe the SbV mechanism in detail in Section 3. In Section 4 we present our implementation of the SbV mechanism in the P2PDP protocol. In Section 5 we evaluate the performance of the proposed mechanism based on some simulation results. Finally, Section 6 presents some concluding remarks.

[1] Some application-level SDPs support both approaches.

2 Related Work

The past few years have witnessed many new research efforts in the area of service discovery for multihop MANETs. Some researchers have focused on extensions to legacy protocols. Examples are Nordbotten *et al.* [7] and their work on service discovery in scatternets (multihop Bluetooth ad hoc networks), and Varshavsky *et al.* [2] and their cross-layer approach to integrating service discovery functionalities within previous routing protocols for MANETs. Such approaches are either platform-specific or inherit some inefficiency from the underlying protocols. Others propose improvements to the broadcasting of service requests in multihop MANETs, such as Konark [8] and GSD [3]. The Konark architecture introduces the concept of 'service gossiping', in which a node can selectively forward both service requests and replies based on cached announcements from other nodes. The efficiency of the Konark approach, however, is highly dependent on caching of advertised service information, thus being inadequate for grid-like computational services. The GSD architecture controls request broadcasts based on the semantic grouping of services as ontology classes, but its efficiency is also dependent on the advertisement and caching of such classes.

Overall, the aforementioned approaches to service discovery in multihop MANETs focus mainly on reducing the amount of packet transmissions related to service requests in such networks. Nevertheless, to the best of our knowledge, there is no other approach that explicitly tackles the specific problem of reply implosions in *purely* query-based SDPs for multihop MANETs.

3 Suppression by Vicinity (SbV)

3.1 Message Fields and Data Structures

We make two main assumptions about the implementation of our SbV mechanism in query-based SDPs.

First, service requests and replies need to convey information that allows the nodes in the MANET to suppress unnecessary replies. More specifically, each request must convey: (i) a unique request identifier (REQID), (ii) the identification of the last node that forwarded the message (HOPID), and (iii) the number of service instances needed by the inquiring node (NUMMAXREPLIES). Similarly, each reply must convey: (i) the REQID matching the one of the corresponding request, and (ii) the identification of the node which the corresponding request was received from (RETPATH).

Note that most of the aforementioned information is readily available from either SDP messages or their encapsulating packets at the link level. More specifically: (i) the REQID field is commonly present in all query-based SDPs we have surveyed so far, (ii) the value of HOPID in service requests and of RETPATH in service replies can be inferred from the source and destination address fields in their encapsulating packets, and (iii) the value of NUMMAXREPLIES in

service requests can be deduced implicitly depending on the service of interest.[2] Therefore, there is virtually no interference of the SbV mechanism in the format of existing SDP messages (see Subsection 3.3).

Our second assumption is that each node in the MANET hosts a local data structure (pendingList) used to control reply suppressions. In addition to reqID, numMaxReplies, and hopID, which are obtained from service requests, each entry of pendingList has a numReplies field (initially set to 0) that records the amount of replies overheard by the node, and an associated timer (cleanUp) that defines the lifetime of this entry in pendingList. Upon reception of a service request, a node records it as a pending request in pendingList before rebroadcasting it to neighboring nodes in the MANET. It is important to note that a rebroadcast service request has its hopID information (*i.e.* the source address field in its encapsulating packet) updated with the identification of the current rebroadcasting node, which allows neighboring nodes to keep track of the path traversed by the request in their local pendingList structures. This information will be used as the *return path* of corresponding replies towards the inquiring node (as explained in the following subsections), thus reducing the additional network load generated by ad hoc routing protocols.

3.2 The Proposed Algorithm

Figure 1 shows the pseudocode of the SbV mechanism as executed by each node as soon as it has received a reply. When a node receives a reply to a request it has previously originated (line 2), the node processes the message and does not forward it further in the MANET. If instead the reply is addressed to an inquiring node other than the receiver, the latter first checks whether there is an entry for the corresponding request in its pendingList (line 7). If so, the receiving node checks whether $N_R < N_M$, where N_R and N_M are (respectively) the values of the numReplies and numMaxReplies fields in the corresponding entry of its pendingList. If $N_R = N_M$, it means enough replies have already been sent towards the inquiring node, so the receiving node suppresses (*i.e.* discards) this reply. Otherwise, the receiving node increments the value of the numReplies field in the corresponding entry of its pendingList. It then compares its own identification with the value of retPath in the reply (line 10). If these values are equal, it means the receiving node is in the return path of the reply and hence can forward the message to the next node in the return path, as indicated by the hopID field in the corresponding entry of its pendingList (line 11).

Figure 2 illustrates the operation of the SbV algorithm. In the figure, only nodes w and z are within y's transmission range. Figure 2(a) shows the initial

[2] Note that all SDPs we have studied so far—with the exception of P2PDP [6] (see Section 4) and the work by Varshavsky *et al.* [2]—do not allow any control on the amount of replies per query nor automatic selection of the most suitable providers. Users must therefore manually select the service instances they are interested in from *all* received replies, possibly leading to bad selection (*e.g.* rashly selecting non-localized providers may increase inter-node interference in the MANET).

```
Require: msg, localID
 1: if firstCopy(msg) then
 2:    if myReply(msg) then
 3:       process(msg)
 4:       return
 5:    end if
 6:    entry ← pendingList[msg.reqID]
 7:    if entry ≠ NULL then
 8:       if entry.N_R < entry.N_M then
 9:          entry.N_R ← entry.N_R + 1
10:          if msg.retPath = localID then
11:             forward(entry.hopID, msg)
12:             return
13:          end if
14:       end if
15:    end if
16:    discard(msg)
17: else
18:    ...{Deal with duplicate replies}
19: end if
```

Fig. 1. SbV pseudocode

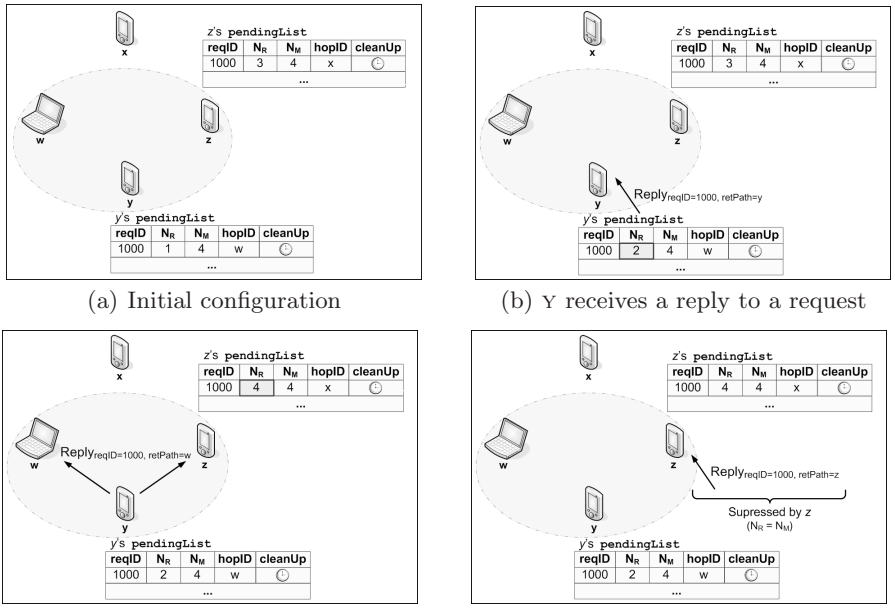
configuration of Z and Y's PENDINGLIST. In Fig. 2(b), Y receives a reply to a request with REQID= 1000, and increments the value N_R of the NUMREPLIES field in the corresponding entry of its PENDINGLIST. As Y is in the return path of the reply (Fig. 2(c)), Y rebroadcasts the message towards W, which is Y's next hop in the return path. Z overhears such rebroadcast and also increments the value N_R of the NUMREPLIES field in the corresponding entry of its PENDINGLIST, but does not in turn rebroadcast that reply because it is not in the reply's return path. In Fig. 2(d), Z receives another reply to the same request, but suppresses such a reply because $N_R = N_M$ in the corresponding entry of its PENDINGLIST.

To summarize, the SbV mechanism reduces the total number of replies conveyed in the MANET by eliminating unnecessary additional replies alongside the return path from replying nodes to the inquiring one. This alleviates the reply implosion problem, which is intrinsic of query-based SDPs.

3.3 Application-Level Forwarding Scheme

Using the SbV mechanism, the service replies are sent towards the inquiring node through application-level forwarding. There are two alternative mappings of this scheme onto the link level: using unicast or broadcast/multicast transmissions.

For link-level unicast mappings, the RETPATH value associated with replies is inferred from the destination address field in the encapsulating packets (*e.g.* the destination MAC address in IEEE 802.11 packets). This address field is filled with the value of the HOPID field in the corresponding entry of PENDINGLIST (which indicates the link-level address of the next node in the return path), as



(a) Initial configuration

(b) Y receives a reply to a request

(c) Y rebroadcasts the reply towards W

(d) Z receives another reply to the request

Fig. 2. Scenario illustrating the SbV mechanism

part of the forward operation (line 11 in Fig. 1). For a participating node to overhear replies from its neighbors, however, its network interface must work in promiscuous mode. Besides the security issues involved, this alternative has the drawback that, in promiscuous mode, the node must process the payload of *all* packets (not only those pertaining to the SDP) at the higher levels, which results in waste of resources (CPU, memory and energy) that are crucial to computational services.

For link-level broadcast/multicast mappings, nodes do not need to work in promiscuous mode; however, the destination link address field in packets encapsulating reply messages do not specify a single recipient, so an additional RETPATH field (with the link-level address of the next node in the return path) is needed in such messages. Further, a statement like $msg.retPath \leftarrow entry.hop\ ID$ must be added as part of the forward operation in Fig. 1; such a statement allows the receiving node to update the reply's RETPATH field with the value of the HOPID field in the corresponding entry of PENDINGLIST, thus allowing the correct node to forward the reply to the inquiring node. As link-level broadcast/multicast mappings consume less computational resources, we have adopted them in our implementation of SbV for the P2PDP protocol.

It is worth noting that for MANETs in which the media access control is based on CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance), broadcast transmissions are less reliable and prone to collisions in comparison with unicast transmissions. This is mainly due to the lack of acknowledgments, RTS/CTS

(Request/ Consent to Send) dialogues, and a mechanism for collision detection. The problem of collisions in link-level broadcast transmissions may be rather alleviated if nodes are prevented from all replying at around the same time. Interestingly, the single-hop version of P2PDP already implements an algorithm in which replies from different collaborators are time-shifted, as discussed in the following section. Regarding the lack of acknowledgments, an implicit acknowledgment mechanism for broadcast transmissions could be used. To understand this, consider again the example of Fig. 2. When W receives the reply message from Y (Fig. 2(c)), being in the return path, it will forward the message. Such a transmission will be overheard by Y (as it is within W's range); Y could then regard this transmission as a higher-level acknowledgement from W. Nonetheless, many subtle issues arise if a retransmission policy based on such implicit acknowledgments is devised to improve the reliability of the discovery protocol. We argue that such additional complexity is not worthwhile, as reply messages are always subject to suppression along the remaining path towards an inquiring node. In fact, the experimental results presented in Section 5.2 demonstrate that, in scenarios of pedestrian mobility, the discovery efficiency in the presence of the SbV algorithm is kept high even without such a retransmission policy.

4 Implementation

We have implemented our SbV mechanism as part of the P2PDP protocol [6]. Along this section, we give a quick overview of the protocol, emphasizing the points where changes were made to accommodate the SbV mechanism.

4.1 Peer-to-Peer Discovery Protocol

Nodes can play two main roles in P2PDP: *collaborators* or *initiators*. Initiators demand computational services from collaborators, which offer their resources—*e.g.* CPU cycles, memory and disk space—for the provisioning of such services. An initiator sends service requests (IREQ messages) to the collaborators and, based on the received replies (CREP messages), define a list containing the collaborators that are more suitable to provide the service. Figure 3 depicts the format of IREQ and CREP messages and illustrates an example of the protocol operation in multihop MANETs.

A collaborator adopts two criteria to decide whether it is able to provide the requested service. The first criterion acts as an admission control, checking weather the collaborator indeed offers the service (*e.g.* if it hosts a specific Web service or a Java virtual machine). The second criterion defines the suitability of the collaborator in providing the service. Crucially, the initiator maps the required service onto the amount of resources needed for its provision. The *context of interest*—indicated in the CTXTINFO field of IREQ messages—allows the initiator to ask collaborators about the desired service, which resources are needed for the service provisioning, and the relative importance among such resources. The initiator also determines in the NUMMAXREPLIES field of IREQ messages

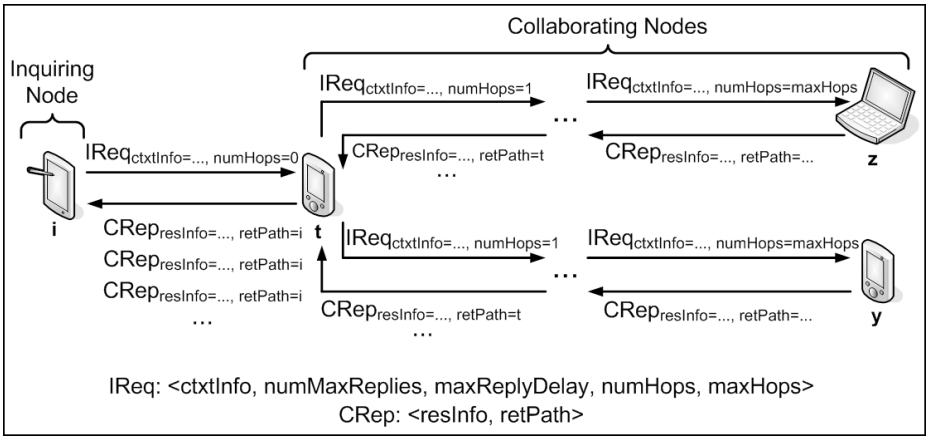


Fig. 3. Example of P2PDP messages

the number of service instances to be involved. Based on such information, a collaborator builds its CRep message, informing in the resInfo field the address of the service (*e.g.* a URL to a Web service or the network-level address of the node), and the resource availability related to the provisioning of such service.

We have introduced new fields in the IReq and CRep messages to allow the operation of P2PDP in multihop MANETs. The numHops and maxHops fields in IReq messages indicate respectively the current and maximum number of hops associated with such messages, and are used to constrain the diameter of service requests. The retPath field in CRep messages is used for forwarding such messages to inquiring nodes, and it is necessary due to the adoption of link-level broadcast transmissions in our application-level forwarding scheme, as discussed in Section 3.3.

4.2 Controlled Delay of CRep Messages

In the P2PDP protocol, each device willing to collaborate with the provision of a particular service delays the transmission of its CRep messages according to a timer. This timer is set to be inversely proportional to the availability of the required resources at the collaborating node. This way, nodes that are more resourceful reply earlier to service requests. If the total number of replies generated in the MANET is larger than the requested maximum number of replies N_M (which is set by the numMaxReplies field in IReq messages), the initiator selects the first N_M received messages as the most suitable replies. When a node receives a request, it gathers its current state in terms of the resources of interest for the given request to compute the reply delay. Importantly, all devices in the MANET must employ the same criterion for such computation. In the implementation of P2PDP, a collaborating nodes sets the reply delay to τ time units as given by

$$\tau = \left(1 - \omega \sum_{i=1}^{N} \left(\frac{\alpha_i P_i}{\sum_{j=1}^{N} P_j} \right) \right) D_{\max} - 2HS, \begin{matrix} 0 \leq \alpha \leq 1 \\ 0 < \omega \leq 1 \end{matrix}, \tag{1}$$

where N represents the number of different resource types the collaborating node should take into account. P_i is the weight that describes the relative importance of each resource type i, $1 \leq i \leq N$. Both N and P_i are described as part of the CTXTINFO field in the request. α_i is the normalized level of current availability (in the interval $[0, 1]$) of resource type i at the collaborating node. $D_{\max}$ is the maximum reply delay, which is also obtained from the request (MAXREPLYDELAY field). H and S are used for considering the transfer delays that IREQ and CREP messages may experience. H is the distance in hops (obtained from the HOPCOUNT field in the IREQ message) between the collaborating node and the inquiring node, and S is a tuning parameter representing the transfer delay at each transmission. Finally, ω indicates the willingness (also in the interval $[0, 1]$) of the collaborating node to participate in the resource provisioning. τ is undefined for $\omega = 0$; such a value means the user is not willing to participate, thus the collaborating node will not send replies. In this case, the node will only act as an intermediate in the message forwarding process.

We highlight that the delay reply mechanism provides a time shift in the transmission of replies, thus allowing for a reduction in the number of collisions of these messages when link-level broadcast transmissions are used.

5 Performance Evaluation

We carried out a set of experiments with the SbV mechanism. These experiments were conducted with two different simulators to evaluate two different aspects of our approach: scalability and discovery efficiency.

5.1 Scalability Analysis

We analyzed the scalability of the SbV mechanism using the ns-2 simulator [9]. All experiments in this simulator consider a fixed node density within the MANET (using topologies with a constant number of nodes within the same transmission range) so the impact of increasing the number of nodes in the MANET could be properly evaluated. The results presented in this section correspond to the average of a hundred sample runs per simulated scenario with a 95% confidence level. This analysis was mainly focused on the evaluation of two metrics: the number of reply messages in the MANET and the suppression diameter of these messages. Table 1 presents the parameters adopted in the simulated scenarios.

The average load of reply messages in the MANET was computed using, for each scenario, the mean number of packets involving these messages. Importantly, this metric also allows us to deduce whether there is a significant reduction in the energy consumption of devices in the MANET due to the suppression of replies, given that transmissions are known to be responsible for a high energy consumption. Using this metric, we compared two purely query-based SDPs: one in which service replies are sent by unicast to inquiring nodes (we called

Table 1. Parameters for ns-2 simulations

Parameter	Value
Number of nodes (N)	10 to 240
Percentage of collaborating nodes (p)	20% to 80%
Maximum number of replies (R)	1 to 10
Node density	5
Distance between nodes	10m

it UCast), and another in which replies are sent through application-level forwarding, with the SbV mechanism incorporated in the forwarding process. In both protocols, the inquiring nodes broadcast service requests by flooding, and no service announcements are employed. Figure 4 presents the number of reply messages as a function of the number of nodes for different percentages of replying devices. The vertical error bars indicate the confidence intervals. The results show that the adoption of the SbV mechanism allows for an increasing reduction—with respect to the UCast protocol—in the total number of transmissions, as the number of devices in the MANET increases. We also observe an even higher level of suppressions when there is a larger percentage of nodes (p) in the MANET with interest in collaborating on service provisioning. These results give a clear idea of the scalability that protocols adopting the SbV mechanism can achieve, such as in our implementation of P2PDP.

The suppression diameter of reply messages measures the distance (in number of hops) between the inquiring node and the nodes where suppressions occurred. This metric allows us to evaluate the degree of distribution of the load alleviation provided by SbV among the nodes in the MANET, and consequently the energy savings among the nodes due to the reduction in the amount of transmissions. Figure 5 presents the distribution of suppressions as a cumulative distribution function (CDF) for different numbers of nodes and percentages of replying nodes. To better illustrate the distribution of suppressions through the MANET, the results presented in Fig. 5 are contrasted with a uniform CDF (represented by the straight line in the figure). We observe a better distribution of suppressions as the number of nodes and the percentage of replying nodes (p) increase. Again, this suggests the scalability of our proposed approach.

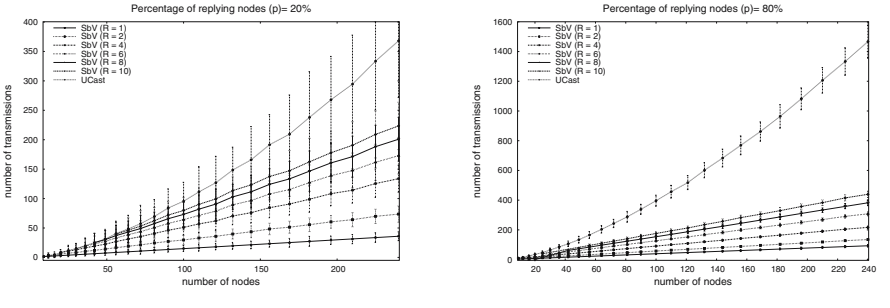


Fig. 4. Network load in the MANET due to reply messages

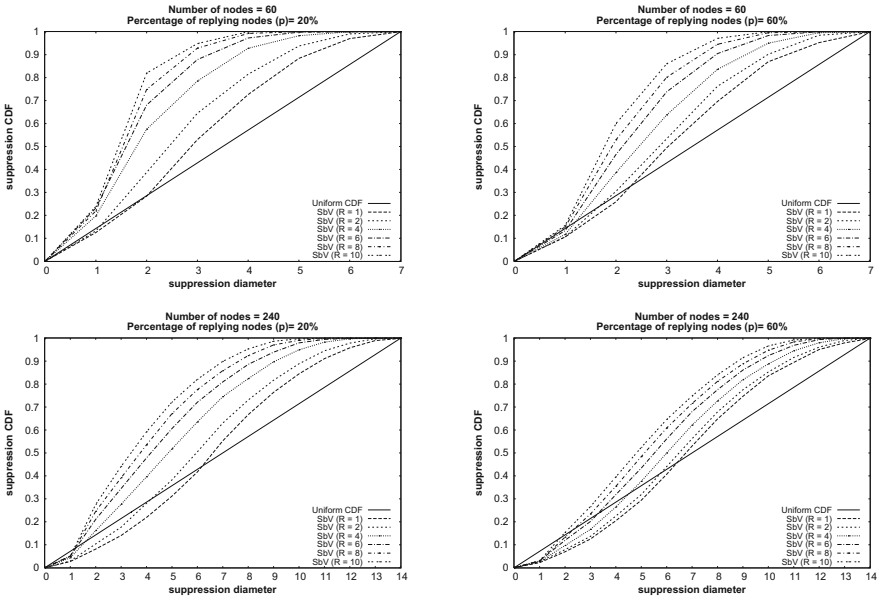


Fig. 5. Distribution of reply suppressions in the MANET

5.2 Discovery Efficiency

To observe the impact of mobility on the efficiency of the P2PDP discovery process using the SbV mechanism, we have implemented a modified version of this protocol for multihop MANETs, as well as a testing application to run on top of it. Both implementations were done in Java, using the CDC (Connected Device Configuration) J2ME profile as our reference platform. Our testing application consisted of a master-worker matrix-matrix multiplication program. For the purposes of our evaluation, we employed a very simple distributed multiplication algorithm: given matrices $\mathbf{A}_{m\times n}$ and $\mathbf{B}_{n\times p}$, a master node computes $\mathbf{C}_{m\times p} = \mathbf{AB}$ by selecting p worker nodes with the P2PDP protocol and sending to each worker node i ($1 \leq i \leq p$) a copy of matrix $\mathbf{A}$ along with matrix $\mathbf{b}^i_{n\times 1}$ (transposed vector whose elements are those of the i-th column of $\mathbf{B}$). Each worker node i computes matrix $\mathbf{c}^i_{n\times 1} = \mathbf{A}\mathbf{b}^i_{n\times 1}$ and returns it to the master node, which then builds each i-th column of $\mathbf{C}$ from $\mathbf{c}^i_{n\times 1}$. The selection of the worker nodes in the MANET that take part in the task is made by only considering those nodes with the most available CPU and memory resources—more specifically, $N = 2$, $P_{\text{CPU}} = 4$, and $P_{\text{mem}} = 1$ in Eq. 1.

We deployed our implementation in the NCTUns simulator and emulator [10]. To do so, we performed some changes to the underlying monitoring service that is part of the original P2PDP implementation. This service[3] is responsible for gathering information about the current state of a mobile node, including

[3] The monitoring service used by P2PDP corresponds to the implementation available at the MoCA architecture [11].

connectivity, CPU load, available energy and memory, and disk storage space. In the NCTUns platform, a single machine runs several (virtual) nodes interconnected by a simulated MANET, but with no kernel isolation between them. Thus, the use of the original monitoring service would lead to unrealistic scenarios in which all nodes in a simulated MANET would have the same state information. To tackle this, we have implemented a "fake" monitoring service that provides randomly generated state information for each different node in a simulated MANET.

The simulation scenarios consisted of 40 nodes placed in an obstacle-free, 500m X 500m area. The initial position of each node was set at random, with the constraint that at the beginning of the simulation the nodes formed a connected topology. The first scenario consisted of a stationary topology. In the remaining scenarios, the movement of nodes followed the random walk model. In such a model, each node moves in a random direction for some seconds—in a speed that is uniformly distributed in the range $]0, S_{max}]$—then chooses a new random direction, with no pause between the direction changes. This corresponds to a worst-case mobility scenario for each speed range. Table 2 summarizes the parameters adopted in the scenarios simulated with the NCTUns platform.

Table 2. Parameters for NCTUns simulations

Parameter	Value
Number of nodes (N)	40
Number of resource providers	10
Maximum number of replies (R)	4
Transmission range	100m
Maximum node speed (S_{max})	0 to 5m/s

The discovery efficiency for each simulation scenario was measured as a sample proportion calculated over 100 runs. Each run consisted of a single resource consumer issuing a single IREQ message to a set of resource providers. The sample proportion indicates the percentage of runs in which the protocol delivered *at least* R replies to the resource consumer, as determined by the NUMMAXREPLIES field in the IREQ message. The number of resource providers at each run was fixed to 10, which corresponds to 25% of the nodes in the simulated scenarios. Such a percentage was chosen based on the study by Hughes *et al.* [12], which states that in Gnutella—a famous P2P, collaboration-based file-sharing system—this percentage of participants is responsible for 98% of all service provisions.

Figure 6 presents the discovery efficiency of the P2PDP protocol extended with the SbV algorithm as a function of the maximum node speed (S_{max}). The vertical error bars correspond to the 95% confidence intervals for each sample proportion. The results show that the protocol behaves well under situations of human mobility (from 0.8 to 1.2m/s).

As it can be observed in Fig. 6, even for the stationary scenario ($S_{max} = 0$) the protocol does not reach 100% efficiency—the sample proportion is 92%, with

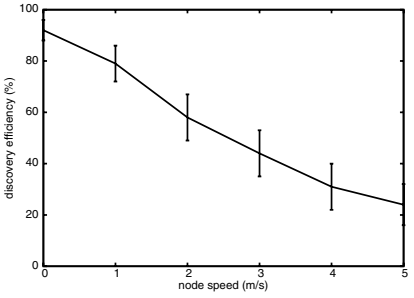


Fig. 6. Discovery efficiency in a mobile scenario

±4.13 confidence intervals. This is due to the drawbacks stated in Section 3.3 regarding the application-level forwarding scheme being mapped onto link-level broadcast transmissions in CSMA/CA enabled nodes.

6 Conclusions

In this paper, we have presented the design and implementation of a mechanism called Suppression by Vicinity (SbV) to reduce the implosion of reply messages in purely query-based SDPs for multihop MANETs. Our experimental results show that the proposed SbV mechanism is efficient in controlling the amount of service replies transmitted in the MANET. Moreover, the additional processing the SbV mechanism generates is well distributed among the nodes. In particular, this prevents greater energy drain rates on nodes nearby the inquiring node, thus promoting an indirect balance on energy consumption due to transmissions. Finally, the SbV mechanism behaves well in the mobile application scenarios we are interested in, which involves pedestrian (walking) mobility.

During the development of this work, some aspects have been identified for future investigation. The first one is the impact of the maxReplyDelay parameter on the efficiency of the SbV mechanism in the P2PDP protocol. Fine-tuning this parameter—*e.g.* as a function of the transmission delay of messages—is essential to reduce the discovery time without increasing the number of reply collisions, which is achieved through the asynchrony in the transmission of these messages. Still in this context, we believe it is important to investigate the influence of clock drifts among different equipment on the timers associated with the SbV mechanism and its implementation on the P2PDP protocol. A second point is that we have considered only low-mobility scenarios in our simulations. In more dynamic scenarios, the concept of return path the SbV algorithm uses for conveying reply messages is likely to reduce the discovery efficiency considerably. To deal with this, we are currently investigating alternative implementations of the SbV mechanism that automatically resort to using traditional ad hoc routing protocols whenever a failure is detected in the return path.

References

1. Marin-Perianu, R.S., Hartel, P., Sholten, H.: A classification of service discovery protocols. Technical Report TR-CTIT-05-25, Centre for Telematics and Information Technology, University of Twente (2005)
2. Varshavsky, A., Reid, B., de Lara, E.: A cross-layer approach to service discovery and selection in MANETs. In: Proceedings of the 2nd IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS) (2005)
3. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: Toward distributed service discovery in pervasive computing environments. IEEE Transactions on Mobile Computing 5(2), 97–112 (2006)
4. Zhu, F., Mutka, M.W., Ni, L.M.: Service discovery in pervasive computing environments. IEEE Pervasive Computing 4(4), 81–90 (2005)
5. McKnight, L.W., Howison, J., Bradner, S.: Wireless grids: Distribute resource sharing by mobile, nomadic, and fixed devices. IEEE Internet Computing 8(4), 24–31 (2004)
6. Lima, L.S., Gomes, A.T.A., Ziviani, A., Endler, M., Soares, L.F.G., Schulze, B.R.: Peer-to-peer resource discovery in mobile grids. In: Proceedings of the 3rd International Workshop on Middleware for Grid Computing (MGC), pp. 1–6. ACM Press, New York (2005)
7. Nordbotten, N.A., Skeie, T., Aakvaag, N.D.: Methods for service discovery in bluetooth scatternets. Computer Communications 27(11), 1087–1096 (2004)
8. Lee, C., Helal, A., Desai, N., Verma, V., Arslan, B.: Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services. IEEE Transactions on Systems, Man and Cybernetics 33(6), 682–696 (2003)
9. Information Sciences Institute: The network simulator ns-2 (1995)
10. Wang, S., Chou, C., Huang, C., Hwang, C., Yang, Z., Chiou, C., Lin, C.: The design and implementation of the NCTUns 1.0 network simulator. Computer Networks 42(2), 175–197 (2003)
11. Sacramento, V., Endler, M., Rubinsztejn, H.K., Lima, L.S., Goncalves, K., Nascimento, F.N., Bueno, G.A.: MoCA: A middleware for developing collaborative applications for mobile users. IEEE Distributed Systems Online 5(10) (2004)
12. Hughes, D., Coulson, G., Walkerdine, J.: Free riding on Gnutella revisited: the bell tolls? IEEE Distributed Systems Online 6(6) (2005)